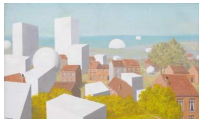





UNIVERSITÉ LIBRE DE BRUXELLES
COMPUTER SCIENCE DEPARTMENT

MEMO-F403 PREPARATORY WORK FOR THE MASTER THESIS

AN EFFICIENT AND PARALLEL
ABSTRACT INTERPRETER IN SCALA
— PREPARATORY WORK —
PRESENTATION



Olivier PIRSON — opi@opimedia.be
 orcid.org/0000-0001-6296-9659

August 25, 2017



VRJE UNIVERSITEIT BRUSSEL



Promotors Coen DE ROOVER
Wolfgang DE MEUTER
Advisor Quentin STIEVENART



An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work
—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

1 Abstract interpretation for static analysis

2 Concrete vs abstract interpretation

3 Parallelism

4 References



Abstract interpretation for static analysis

An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

Abstract interpretation is a technique to perform static analysis

A *static analysis* of a program is made without executing the program

The goal is to *prove* the correctness (or other property) of the program



An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

1 Abstract interpretation for static analysis

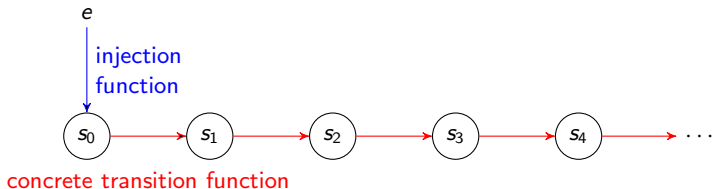
2 Concrete vs abstract interpretation

3 Parallelism

4 References

Concrete interpretation

trace: concrete interpretation with *small-step semantics*, for one instance



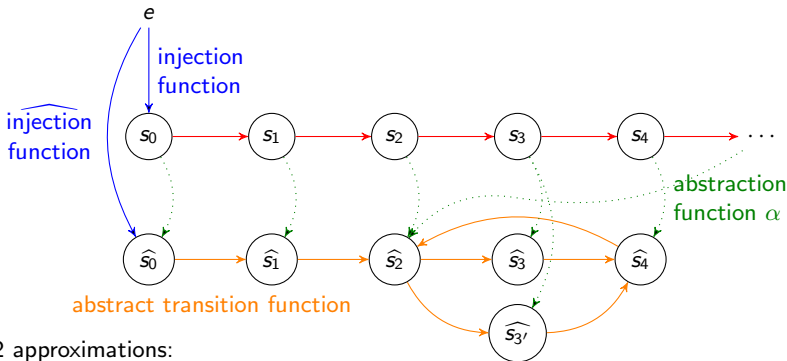
A program is executed by interpreter = *Abstract Machine* (AM)

- e is for expression, i.e. a program
- one execution is for one instance on this program
- s_i are states during this execution



Abstract interpretation

trace: concrete interpretation with *small-step semantics*, for one instance



2 approximations:

- finite state space
- abstract transition function returns all directly reachable states

state graph: abstract interpretation, for *all* instances

Abstracting Abstract Machine (AAM)

"The *abstract simulates the concrete*" (Matthew MIGHT)



Concrete interpretation

An Efficient and
Parallel Abstract
Interpreter in Scala

—
Preparatory
Work

—
Presentation

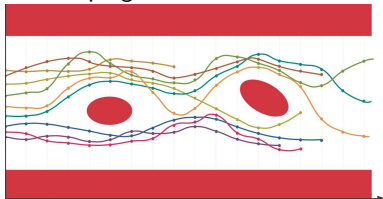
Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

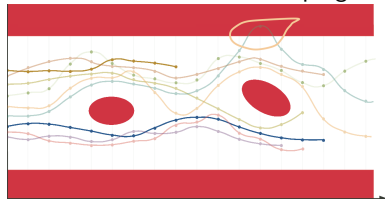
Parallelism

References

Correct program



Incorrect program



We want to prove the correctness of the program

Testing:

- is not enough to prove
- exhaustivity is impossible



Abstract interpretation

An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

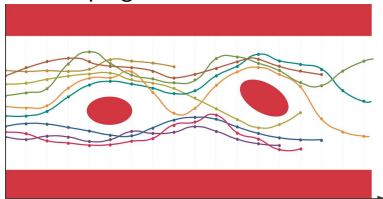
Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

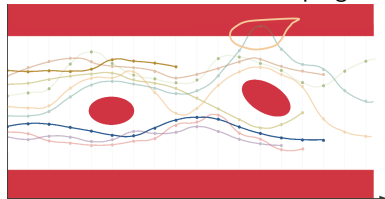
Parallelism

References

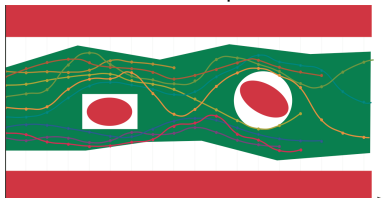
Correct program



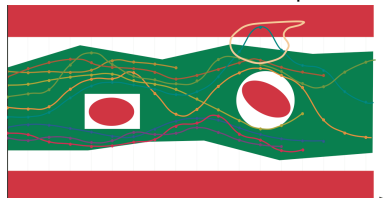
Incorrect program



Correct abstract interpretation



Incorrect abstract interpretation





Abstract interpretation

An Efficient and
Parallel Abstract
Interpreter in Scala

—
Preparatory
Work

—
Presentation

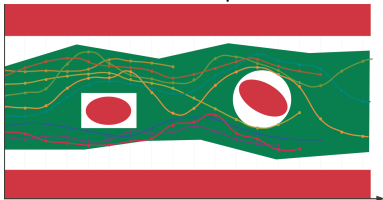
Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

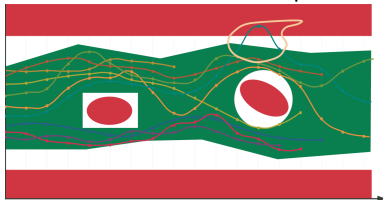
Parallelism

References

Correct abstract interpretation



Incorrect abstract interpretation



- Decidability?
- Soundness vs completeness?
- Good complexity vs precision?

⇒ parallelism to improve speed



An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

1 Abstract interpretation for static analysis

2 Concrete vs abstract interpretation

3 **Parallelism**

4 References



Goal of the future master thesis: parallelism

An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

Parallelism to have good precision and speed

Choices for the future master thesis:

- Actor model: model of concurrency with
 - no shared resources
 - exchanges of messages
- Akka: framework Scala

Time is up!



An Efficient and
Parallel Abstract
Interpreter
in Scala

—
Preparatory
Work

—
Presentation

Abstract
interpretation for
static analysis

Concrete vs
abstract
interpretation

Parallelism

References

- 1 Abstract interpretation for static analysis
- 2 Concrete vs abstract interpretation
- 3 Parallelism
- 4 References



Thank you!

References:

- All documents and \LaTeX sources:

<https://bitbucket.org/OPiMedia/efficient-parallel-abstract-interpreter-in-scala-preparatory>

- René MAGRITTE, *Le Calcul Mental*, 1940

<http://www.artnet.com/artists/ren%C3%A9-magritte/le-calcul-mental-oU6yWQzE-ERgxZSTgANE-g2>